Adaptive Computer Systems

Russell Abbott, Ontario Engineering International Inc., 3333 Harrison St #6, Riverside, CA 92503, PH 909 283 5971,
rabbott@oeico.com
Scott Cannon, Utah State University, Logan, Utah

Abstract.— A scaleable Plug and Play interface has been developed and demonstrated in a multibus architecture with 4-8 modules. The modules can be added and removed at will without any reprogramming or rebooting of the modules. Any failure or latchup of any network module will not stop bus communication with the remaining modules. A demonstration has been conducted simulating the addition, removal, and failure of various nodes.

This Plug and Play interface will be capable of supporting an adaptive network architecture that can be implemented into future spacecraft or robotic outposts on Mars. When the Plug and Play architecture is combined with a "standardized electrical and mechanical interface" the ability for spacecraft to reconfigure themselves is gained allowing replacement, upgraded or adhoc subsystems to be added to the spacecraft for new missions.

The implementation of this type of Plug and Play system will be to lower barriers of entry to providers of spacecraft components and subsystems by providing a common interface that eliminates the problem of proprietary bus architectures. The benefits of a Plug and Play interface will enable faster spacecraft integration thus reducing time and cost. Heritage subsystems can be equipped with front end Plug and Play modules to interface with any spacecraft reducing development costs. Dynamic reconfiguration will enable on-orbit assets to be updated on a continuous basis by using low cost launch vehicle systems versus sending up new spacecraft.

TABLE OF CONTENTS

1.0 INTRODUCTION

Requirements for Space Qualified Plug and Play Architecture

Computer and embedded system development continues to be burdened by divergent requirements. On one hand the performance must increase at a nearly logarithmic rate, while on the other hand the system cost must stay the same or decrease. Several applications, such as those found in [1]spacecraft infrastructure equipment, are also burdened with increasing capabilities while decreasing the board size and ultimately the volume, which the unit(s) occupy and the need for radiation-hardened technology.

The connection fabric between microprocessors and peripherals has traditionally been comprised of a hierarchy of single and multi-drop buses (Figure 1). For the multi-drop bus, as the number of devices added to the bus increases, the bandwidth drops proportionately. Devices are placed at the appropriate level in the hierarchy according to the performance level they require. Low performance devices are placed on lower performance buses which are bridged to the higher performance buses so as not to burden higher performance devices. Bridging is also done to address legacy interfaces.

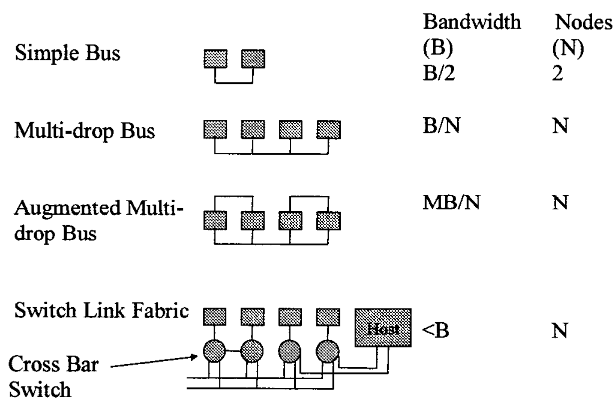

Figure 1 - System Topology Trends. The Switched Link Fabric with a bus directing Cross Bar Switch maximizes the utilization of the available system bandwidth while reducing latency losses.

Over the past several years the shared multi-drop bus has been exploited to its full potential by augmenting with local bus structures. Many techniques have been applied, such as increasing frequency, widening the interface, pipelining transactions, splitting transactions, and allowing out of order completion. Continuing to work with a bus in this manner creates several design issues. Increasing bus width reduces the maximum achievable frequency due to skew between signals. More signals will also result in more pins on a device, resulting in higher product cost and a reduction in the number of interfaces the device can provide.

To address the needs of future space systems an embedded system component network architecture utilizing a switched link fabric is proposed. The architecture is for a point to point, moderately parallel, packet-based interconnect. In embedded systems applications, it must be limited to no impact on the software infrastructure that operates over it.

[1] 0-7803-7231-X/01/$10.00/© 2002 IEEE
[1] IEEEAC paper #477, Updated Nov 12, 2001

Report Documentation Page				*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **2002**	2. REPORT TYPE	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Adaptive Computer Systems**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) **Russell Abbott; Scott Cannon**	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Ontario Engineering International, Inc,3333 Harrsion St #6,Riverside,CA,92503**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
A Scaleable Plug and Play interface has been developed and demonstrated in a multibus architecture with 4-8 modules. The modules can be added and removed at will without any reprogramming or rebooting of the modules. Any failure or latchup of any network module will not stop bus communication with the remaining modules. A demonstration has been conducted simulating the addition, removal, and failure of various nodes. This Plug and Play interface will be capable of supporting an adaptive network architecture that can be implemented into future spacecraft or robotic outposts on Mars. When the Plug and Play architecture is combined with a "standardized electrical and mechanical interface" the abilitiy for spacecraft to reconfigure themselves is gained allowing replacement, upgraded or adhoc subsystems to be added to the spacecraft for new missions. The implementation of this type of Plug and Play system will be to lower barriers of entry to providers of spacecraft components and subsystems by providing a common interface that eliminates the problem of proprietary bus architectures.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**7**	

A Plug and Play interface is implemented in each subsystem, major components including cable harnesses, structural elements, propulsion systems and Orbital Replacement Units (ORU). The interface components will be radiation-hardened by design to insure minimal weight and small size requirements.

Plug-and-Play Concept Definition

Our vision in this effort is a spacecraft built of composite panels incorporating electronic systems as an integral part of the panel assembly. The spacecraft would be constructed by plugging together multiple panels through standardized connectors. The connectors would be capable of configurable routing to permit appropriate mating of paired signals among the electronic systems on connected panels. This concept will allow a spacecraft to be easily assembled and modified, as well as easily repaired and reconfigured once the spacecraft is on-orbit. Achieving this vision will require the development of a switch linked fabric architecture linking all the subsystems and replaceable structural elements that comprise a spacecraft together. A scalable Plug and Play interface has been developed and demonstrated with 3 to 8 modules. These modules can be added and removed at will without any reprogramming. They can be mounted onto a composite panel structure representative of micro-spacecraft incorporating integral cable, electrical shielding and thermal management structures developed by OEI.

In future development efforts, a Plug and Play interface will be developed in the form of a multi-port Cross Bar Switch capable of supporting the switch link fabric architecture that can be implemented into future reconfigurable spacecraft. This effort would include the fabrication of a simulated cube shaped micro-spacecraft with eight Plug and Play panels to demonstrate the capability of this interface. These panels will represent various subsystems typically found onboard spacecraft. The micro-spacecraft structure will include embedded cabling capable of reconfiguring itself to allow different subsystems to be added to the spacecraft. OEI has already developed the structure, panels and cabling interfaces for this cube shaped micro-spacecraft through previous research and development efforts.

This system will be promoted as a means to lower barriers of entry to providers of spacecraft components and subsystems by providing a common interface that eliminates the problem of proprietary bus architectures.

Plug and Play Concept Implementation

To be effective, the envisioned Plug-and-Play concept must be reduced to a set of elements that can be integrated to implement a self-configuring structure. The needed elements include: (1) interconnect/packaging to provide structural support, interconnect, thermal management, and perhaps radiation shielding, (2) electronics to perform the configurable routing functions and a foundation for protocol implementation, and (3) software to manage the configuration and implement the protocols. Each of these elements is treated in the following subsections.

Interconnect/Packaging

Current spacecraft are constructed using subsystems from various manufacturers with unique interconnection requirements. Typically most of these subsystems use cables to connect to the spacecraft bus. This provides some flexibility to the spacecraft integrator as they can define the cable connection to the rest of the spacecraft. Unfortunately this incurs a large weight penalty and makes the implementation of integrating the structure and wiring very difficult when used on different spacecraft and extremely difficult for Orbital Replacement Units (ORU). This means any attempt at developing an ORU must account for the high pin count across the interface to accommodate the widest possible number of interfaces that can be expected. This has the effect of driving the size of the connector and increasing the weight unnecessarily.

There are three generic types of interconnect interfaces that must be addressed in defining a PnP (Plug and Play) concept. They are (1) power interfaces, (2) analog signal interfaces, and (3) digital signal interfaces. Each has its own unique requirements. While the ultimate goal of PnP is to permit any generic interconnect to occupy any location in the connection matrix, the near term implementations will probably need to geographically segregate the types if a self-organizing capability is to be retained.

Power interconnections may be the most difficult to treat in a general sense. For example, a fundamental decision will be required on whether to just distribute the 28 volt power bus among the modules or to distribute regulated power at the voltages needed for the systems. Our current thinking is that 28 volt distribution is the best alternative with appropriate regulation and voltage generation occurring in the electronic systems located on the panel. The interconnect and routing for the power and ground (return) would have to be sufficiently robust to carry the required current without excessive voltage drop or heating. Consequently, we currently believe that power should be segregated to a section of the connector specifically designed for efficient distribution.

Analog signal interconnections cover a very broad spectrum of current and voltage amplitudes, frequencies, loads, and source impedances. They are likely to be quite susceptible to noise especially if routed near high frequency digital signals. Whenever possible, analog to digital conversion (ADC) should take place within a panel and the information passed digitally to other panels. The analog signal could be reconstructed with a DAC (digital to analog converter) if an analog signal is needed. However, there will likely be some analog signals that must be exchanged among panels. We currently believe that there should be a dedicated segment of the connector devoted to analog.

Digital signal connections also have some complexity associated with them. In the past, most digital signals were single ended and were based on 5 volt technology. So in many cases, devices from different logic families, such as CMOS and TTL, could talk to each other as long as attention was given to fanout, loading, pull-up/pull-down resistors, etc. However, as digital technologies move to smaller feature sizes, the operating voltage is reduced. The result is devices with output logic swings of 0 –5 volt, 0-3.3 volt, 0-2.5 volt, and 0-1.8 volt. Furthermore, many of the high performance devices use differential I/O such as the LVDS (low voltage differential signal) standard.

The significance of this complexity in digital signals will depend on the method used to route the connecting signals from different panels. If purely, mechanical connections are used such as hardwire or relays, the differences in digital signals are of little consequence. The connector simply matches the mating lines and the compatibility of the signals is the responsibility of the designers of the subsystems between the two panels. However, if electronic switching (e.g., a cross bar function implemented in an ASIC or an FPGA) is used to perform the routing, the signals must be buffered to conform to the I/O requirements of the electronic switch. In that case, the digital signals might be segregated into single ended and differential sections. The single ended signals might require buffering on the panel to standardize the panel I/O to a single voltage swing.

Structural Consideration of the PnP Concept

The hardware/interface requirement for spacecraft compatible PnP electronics needs to address the following issues: PnP integrated structural modules are one more step in enabling spacecraft structures to become part of the electronics subsystem. The purpose is to reduce the duplication of packaging efforts between the structure and electronics package to realize mass and volume savings. OEI has developed a method to integrate structure, thermal management, electrical interconnection and electronic subsystem packaging into an integrated structure. This is achieved by using either honeycomb or compression molded composite panels as the main support structure. The thermal management system is a passive system using high thermal conductivity materials such as TC1050 for planar configurations and high thermally conductive graphite and fiber materials in an epoxy matrix to form three dimensional thermal paths between the interior and exterior of the spacecraft. This allows connection to the radiator panels or heat pipes going to the radiator panels. These features can then be over molded or laminated with composite materials to form compact structures containing electrical and/or radiation shielding as well as thermal and electrical cabling in the structure. Or the use of honeycomb panels can be used as well if only structural loads are considered or if a heritage spacecraft design is used.

The electrical cabling in the form of flex cables is used to provide the maximum flexibility in the constructing integrated structures. It also allows the creation of multiple bus paths to connect the various subsystems on and between other panels. One must think of the spacecraft structure and cabling as an extension of the electronic subsystem. Think of the subsystems as components on a three dimensional printed wiring board. OEI has developed a method using a single piece connector to connect the electrical cables across the panels to the subsystems without using solder connections or traditional 2 piece connector solutions. This allows the panel and attached subsystems to be physically treated as a Plug and Play modules.

Interconnect modules will be used with heritage subsystems to interface with the panels. These modules will contain the PnP electronics and the connectors to the subsystem and the connector interface to the panel. This allows different connectors to be used with existing subsystems. The interconnect module in its simplest form will contain only a passive hardwired switch to direct the signals to the desired bus in an augmented multidrop bus system. If active bus switching is desired to implement a switch link fabric, then either a semi-active or an electric cross bar switch is required to allow bus switching. The module physical format is based on a standalone module for heritage type subsystems that is attached to the panel or the exterior of the subsystem. Or it can be included in the subsystem that has spare slots or is a new design.

This module will meet the scalable interface requirements between legacy electronic subsystem, new subsystems, integrated structures, ORU's and docking ports. In addition, this system will also enable the lowering of barriers to entry for suppliers of spacecraft components and subsystems. As the supplier no longer has to redesign their product to meet the interface needs of a proprietary bus architecture.

2.0 RESEARCH WORK CARRIED OUT

Plug & Play Software

A Plug & Play network using PC nodes and a common front panel. The front panel will provide network ports for RS232 serial cables from PC network nodes. Network software will run as a protocol layer below demonstration applications software on each PC to support Plug & Play network features and peer-to-peer communications. The network will be implemented within the panel using redundant half-duplex multi-drop busses constructed with RS422 converters as shown in Figure 2.

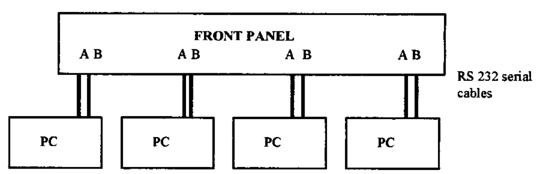

Figure 2 Plug and Play Demonstration Scheme.
Each PC is a laptop with two PCMCIA cards for Bus A & B

The front panel will provide connectors for up to 8 network ports. This panel will appear to be an intelligent switch system that allows a node connected to any port to be aware of all other network nodes and address messages to or receive messages from these other network nodes.

Node-to-panel communications will be standard RS232 serial ports. A node will be added to the network simply by connecting at least one serial cable to the front panel. One serial cable connection will allow support for Plug & Play peer-to-peer communications. Two serial cable connections from a PC node to the front will allow additional demonstration of network fault-tolerance. A node can be removed from the network by disconnecting all serial cables from that node.

When a node is connected to the network through the front panel, the rest of the network will automatically become aware of the identity of the new node. The new node will also be given access to the identities of all other existing network nodes. As long as at least one serial communications line is operating between the front panel and the node, the node will be configured into and recognized by the network.

To demonstrate the self-organization of the Plug & Play network as well as peer-to-peer communications, PCs will be utilized as network nodes. Each PC will display a dynamic interface showing network status, traffic to/from that node, and node processing. Each PC node will be programmed to periodically request processing help from another network. Each PC node will be programmed to accept processing requests and data from the network, perform the indicated processing, and return the results to the requesting node.

Network Design

There are several networking strategies that could be employed for peer-to-peer communications in a dynamically configuring Plug & Play network; from point-to-point connections using software routing or hardware switching to full and half-duplex busses. For robustness in design, efficiency of bandwidth, eliminating single points of failure, and simplicity in meeting a variety of node interfacing requirements and standards, we have chosen a redundant half-duplex multi-drop bus architecture.

The Plug and Play design will consist of the following modules:

Front Panel

The panel will convert each serial port connection to RS 422. One serial communications line (connector) from each port will then be connected to one of two half-duplex multi-drop busses using the RS 232 RTS signal for a Transmission Enable (TxEn) control line according to the diagram in Figure 3.

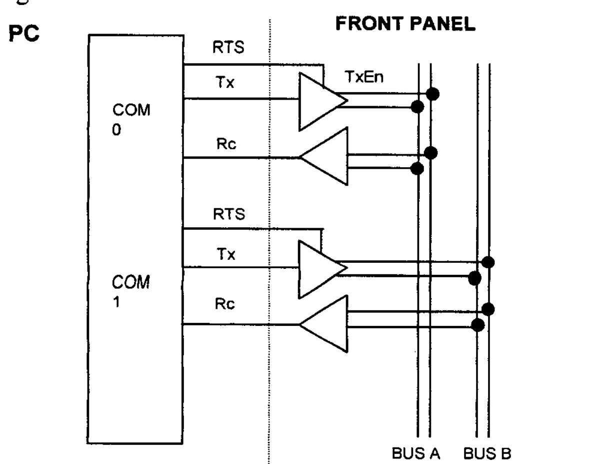

Figure 3 Schematic of Plug and Play Demonstration.

As such, the front panel configures the network into redundant or augmented half-duplex multi-drop bus model symbolically represented in Figure 4. In this model, two communications connections can exist simultaneously (Bus A and Bus B) or one bus can be viewed as a backup for the other. The half-duplex bus was chosen over a full-duplex bus for its electrical interfacing simplicity and robustness. In addition, bus bandwidth is more effectively used at high communications as collisions are avoided.

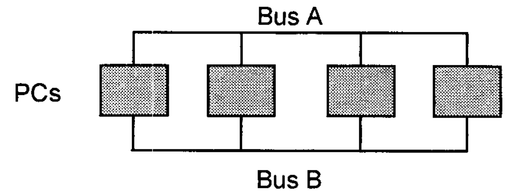

Figure 4 Demonstration implements augmented multidrop bus currently in use.

Network Architecture Protocol

To avoid bus conflicts and provide high-level messaging services to applications software associated with each node, a network software layer will be developed to run on each PC node. All network communications will be performed through this network software layer. This approach insulates application software from network details and protocols. It also allows applications software to be architecture independent, see Figure 5. The network software layer will support multiple cooperative or independent applications.

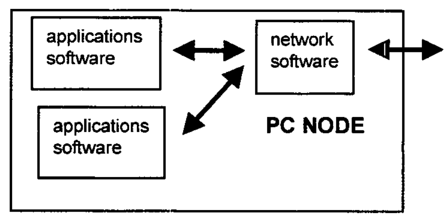

Figure 5 The network software handles digital signals is just one layer of a multilayer approach to implement P&P. Other layers handle power, ground and analog signals

Managing half-duplex bus communications in a dynamic network is most robustly done using a master arbitrator to prevent message collisions. The Plug & Play design implements a *dynamic* master by allowing any one of the processor (PC) nodes to serve in this capacity. The network layer on each node may operate in either master or slave mode. The master node maintains a list of current network slaves (including itself) which it polls in a round-robin fashion. Each slave node in turn is polled to pass communications privileges around the current network. A slave node may only transmit after being polled by the master.

The redundant busses (A and B) eliminate single points of failure within the network panel. All nodes (i.e., network software layers) monitor both communications ports. A network message or poll may arrive on either port. A network message must be transmitted on the polled port. The master initially assumes all nodes communicate on bus A. If a node fails to respond to a poll on bus A, the master will attempt a poll on bus B. If a node then responds on bus B, future communications with that node will be done using bus B. If the node again fails to respond on bus B, the node is considered to have failed (see below).

To implement a full Plug & Play capability and peer-to-peer communications, two situations must be resolved in a shared half-duplex bus design with arbitration: master election and new slave recognition.

1. Master election
Each node is hard-wired with a unique ID. The following describes the actions of the network software layers in cooperating to choose a new master: Since the master is regularly polling all slaves (including itself), each slave can easily detect when the master has failed or otherwise is not operational within a deterministic timeout of bus traffic. When this occurs, each slave immediately idles for a time proportional to its ID. When a node awakens, it checks for a message that may have arrived while it was idle. If no message is present in the incoming queue, the node assumes it is the first to awaken and assumes the role of the new master. It immediately broadcasts an appropriate message to this effect (on both busses).

If an awakening node finds a message has arrived during its idle time, it recognizes the sender of this message as the new master and resumes the network protocol. This same mechanism for choosing a new master also occurs when the network is first powered up. This protocol is a coup variation of more common bus election algorithms.

2. Slave recognition
The master maintains a list of current slave nodes that are polled in a round-robin fashion. If a poll to an existing slave times out (both busses), the slave is assumed to have failed and is removed from the active list. The master broadcasts a message to other nodes to notify them of this failure. Periodically, the current master polls all possible slave IDs to determine if a new slave node has recently joined the network and should be added to the current poll list.

Peer-to-peer applications messaging

An application begins by registering its chosen application name with the network layer. This name consists of 4 bytes (usually ASCII text). The name is arbitrary, but must be unique among the applications currently active in the network.

Network layer messaging— An applications message is sent to the network layer where a network header is prepended and the message is placed on the end of a network message queue to be broadcast over the bus in response to a master poll. When the addressed network layer receives the message it is placed into an incoming queue for the appropriate application. Broadcast messages received by a network layer that are addressed to another node are discarded.

Summary of network protocol— Under normal operating conditions, one node serves as the master node and all others are slaves. Any node's network layer software can operate in either slave or master modes. Since communication occurs over a half-duplex bus, only one node can transmit at any given time, therefore a slave does not transmit on a bus until it is given permission via a poll from the master. The master maintains a list of all active nodes and polls each node on the list in turn.

All network software layers monitor either communications busses or ports. A network message or poll may arrive on either bus. A network message must be transmitted on the polled bus. The master initially assumes all nodes communicate on bus A. If a node fails to respond to a poll on bus A, the master will attempt a poll on bus B. If a node then responds on bus B, future communications with that node will be done using bus B. If the node again fails to respond on bus B, the node is considered to have failed.

Polling— The master polls slaves in the following manner: The ID of the next node to poll is retrieved from the Poll List, a poll message is constructed and addressed to that node, and the message is transmitted on the bus. The Master then delays for a fixed amount of time before checking for a reply from the most recently polled node. After dealing with the reply (or lack thereof) as described below, the master gets the ID of the next active node from the poll list and repeats the process. The master's ID number is also kept in the Poll List, and the master polls itself and replies as if it were also a slave.

Replying to a Poll—All slaves listen to the busses continuously. When a poll message arrives, the slave checks whether the message is addressed to it, and if so the slave sends a reply. Slave replies can either be empty or they can contain an application message. When replying the slave first attempts to retrieve an application message from its outgoing message queue, and if there is no message there it creates a default empty reply.

Slave Failure or Removal—If a slave node fails and therefore does not reply to its next poll, it will be removed from the poll list and the remaining nodes will continue operating normally. The failed or removed node is not polled again until the poll list is reset (see below). The master then broadcasts a notice to all other slaves that the node is now inactive

Adding Slaves to the Network (Resetting the Poll List) Periodically, the master node resets the poll list by adding all possible node ID numbers to the list. At this time any active node that was not previously in the poll list will be polled and can begin communicating on the network by replying to the poll. Nodes that are not active or do not exist on the bus will not reply and will be removed from the list.

Master Failure or Removal—Each slave maintains a timer that is reset every time it detects activity on the busses. If the master stops functioning, the busses are idle since slaves must wait for the master to poll them before they can transmit. When the busses time out, a coup is initiated, which results in the selection of a new master. Since the nodes cannot communicate their ID numbers without a master to organize network communication, the node with the lowest ID number is determined through a process in which each node sleeps (idles operation) for a period of time proportional to its ID number. The first node to wake up transmits a coup message to all nodes and becomes the master. All other nodes find the coup message when they awake. After waking briefly, all nodes (including the new master) idle again for a period of time inversely proportional to their ID number. The result is that all nodes complete the coup at approximately the same time, and network operation resumes with a new master.

Master Conflicts— A network may at times have more than one master. This can occur if two sub-networks (one of which may contain a single node) are combined or if the master is temporarily halted, then resumes operation after a coup has been held and a new master selected. When this occurs, one master will detect the other (usually by receiving a poll message from the other master) and will step down and become a slave. Occasionally, two masters will detect each other at roughly the same time and both step down, leaving the network without a master. When this occurs a new master will be selected via a coup (see Master Failure).

Summary of peer-to-peer (application) communications— Each time an application begins, it registers a chosen name with the local network layer software. The network layer maintains a list of registered applications (local and network). An application may request a copy of this list to determine the names of other current applications in the network. An application may transmit a message to another application using the registered name of the destination application. The network layer broadcasts the message with the appropriate header (in response to a poll from the current master) which is received and queued by the destination network layer. The destination application can read queued messages from its local network layer. A peer-to-peer message is discarded if the destination application is not registered with the local network layer.

If a node fails or is removed from the network, the master transmits a notice to all network layers. This results in all applications currently registered on the removed node to be de-registered. If an application is subsequently re-started on another node, it simply re-registers with the local node.

Applications

Packaging of an active Cross Bar Switch will require the use of an interface module to be added to any system requiring access to the spacecraft bus, see Figure 6. The interface module will have a standard bus interface thereby eliminating the problem of differing connector and pin assignments. This solves the problem how do you integrate heritage and new space subsystems into new spacecraft. A common module design using a multiport device, and an FPGA would be required.

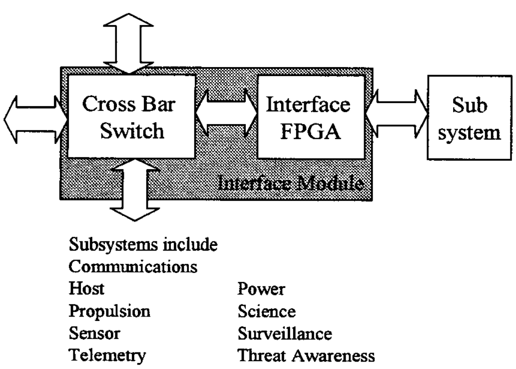

Figure 6 A Plug and Play Interface Module can be added to a variety of subsystems to insure high-speed communications between other subsystems.

3.0 Conclusion

The need for upgradable/reconfigurable electronic systems for spacecraft is apparent, as access to space is still represents a high entry barrier to most companies. The development of a Plug and Play Subsystem Interface will enable increased bus bandwidth, use of heritage subsystems without resorting to proprietary bus architectures. As a side benefit it also allows the bus to become damage tolerant as the Plug and Play Modules can switch busses as required if communication is lost or if the bus traffic exceeds established threshold levels.

4.0 References

Bangnan Xu and Bernhard Walke, "Design issues of self-organizing broadband wireless networks", *Computer Networks, 73-81*, 37(2001)

Koji Nakano and Stephan Olariu, "Randomized Initialization Protocols for Ad Hoc Networks", *IEEE Transactions on parallel and distributed systems*, July 2000

Shlomi Dolev and Jennifer L. Welch, "Crash Resilient Communication in dynamic networks", *IEEE Transactions on computers*, January 1997

Gurdip Singh, "Leader election in the presence of link failures", *IEEE Transactions on parallel and distributed systems,* March 1996

V. A. Koloskov and V. S. Titov, "Self-organization of a fail-proof multimicrocontroller network", *Automation and remote control,* March 1998

Acknowledgements

This effort was sponsored by a Small Business Innovative Research (SBIR) grant under the Air Force Research Laboratory / Space Vehicles Directorate, Kirtland AFB, NM.

***Russell Abbott** is a Technology Director in mechanical and electronic packaging development of high-density interconnection technology. He has developed and led development of high-density interconnection systems at General Dynamics, Hughes Aircraft Co., Raytheon and Ontario Engineering International. He is the He is the Director of Technology for Ontario Engineering International, where high-density interconnection systems and self-engaging connectors are developed and supported. He has a BSME from California Stature Polytechnic University, Pomona California*